

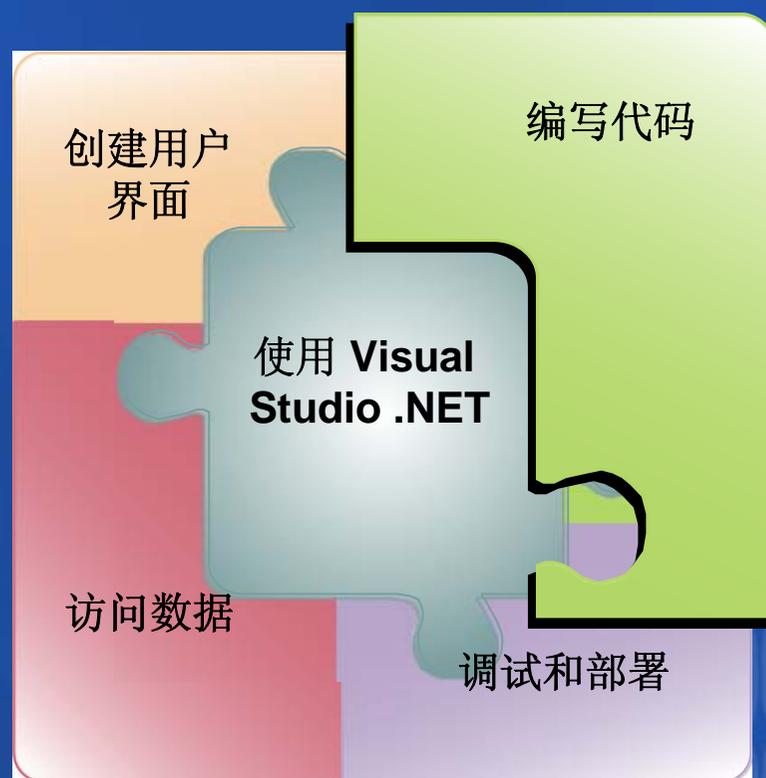
您的潜力，我们的动力

Microsoft[®]
微软(中国)有限公司

课程主题：面向对象设计

讲师：蒋林峰

处理错误和异常 (补)



- 错误类型
- 使用调试器
- 处理异常

您的潜力，我们的动力

Microsoft[®]
微软(中国)有限公司

错误类型

- 语法错误
- 运行时错误
- 逻辑错误

语法错误

语法错误

```
Public Sub PerformAction()  
    Dim newValue As xyz  
    MessageBox.Show("Test")  
End Sub  
End Class
```

语法错误

Name 'MessageBox' is not declared.

从任务列表中查看错误

任务列表 - 已显示 (筛选) 3 个 生成过程错误 任务

!	说明	文件
	单击此处添加新任务	
!	"If" 必须以匹配的 "End If" 结束。	C:\Documents and ... \Form1.vb
!	名称 "a" 未声明。	C:\Documents and ... \Form1.vb
!	需要表达式。	C:\Documents and ... \Form1.vb

运行时错误

Speed = Miles/Hours

‘ 如果 *Hours* = 0, 那么虽然从语句上来讲这是正确的, 但除以0是非法的运算

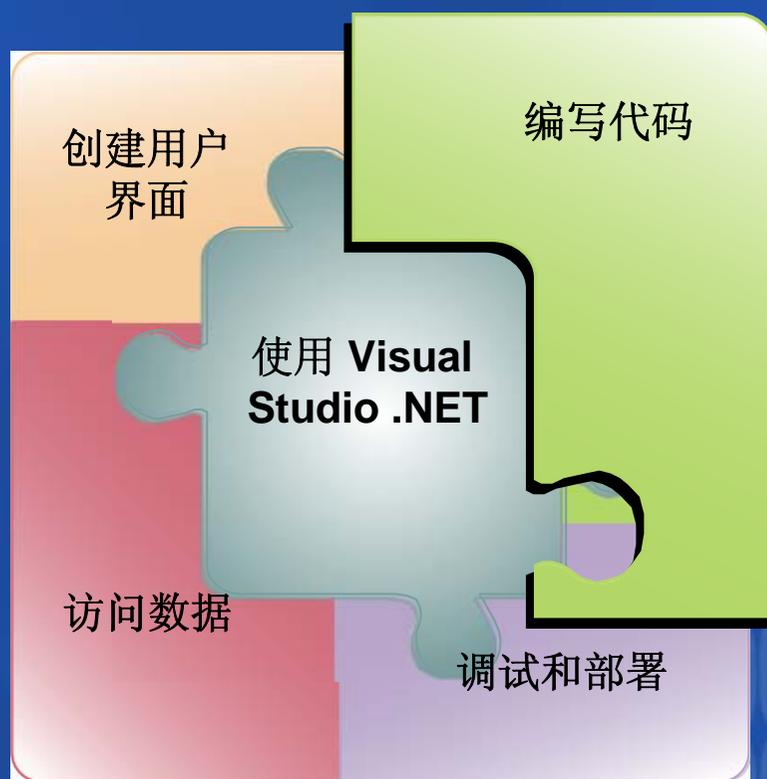


逻辑错误

- 定义: 导致程序产生不正确结果的错误
 - 编译器不能捕获运行时错误
 - 通过测试应用程序和分析运行结果来捕捉

```
Dim x As Integer = 2
Do While x < 10
    ' Code statements
    x -= 1
Loop
```

处理错误和异常



- 错误类型
- 使用调试器
- 处理异常

调试器的使用

- 中断模式
- 使用断点
- 修改断点
- 调试工具栏
- 单步执行代码
- 使用调试窗口
- 使用命令窗口

中断模式

- 中断程序运行
- 在中断模式下，你能够：
 - 逐行执行代码
 - 确定在程序运行时调用的过程
 - 监视变量、属性和表达式的值
 - 使用调试窗口修改变量和属性的值
 - 改变程序流程
 - 运行代码语句

使用断点

- 断点是一种设在代码中的标记, 它可以使 VB 程序执行到某一指定行时暂停执行代码
- 不能在一条不可执行的语句上设置断点

断点

```
Private Sub Button1_Click(ByVal sender As System.Object, _  
    ByVal e As System.EventArgs) Handles Button1.Click  
    Dim maxLen As Integer = 0  
    Dim longestName As String = ""  
    Dim index As Integer  
  
    For index = 0 To names.GetLength(0)  
        If names(index).Length > maxLen Then  
            longestName = names(index)  
        End If  
    Next  
  
    MsgBox("Longest name is " & longestName)  
  
End Sub  
End Class
```

修改断点

新断点

函数 | 文件 | 地址 | 数据

程序执行到函数的该位置时, 中断执行。

函数 (F):

行 (M): 语言 (L):

字符 (R):

条件 (C)... (无条件)

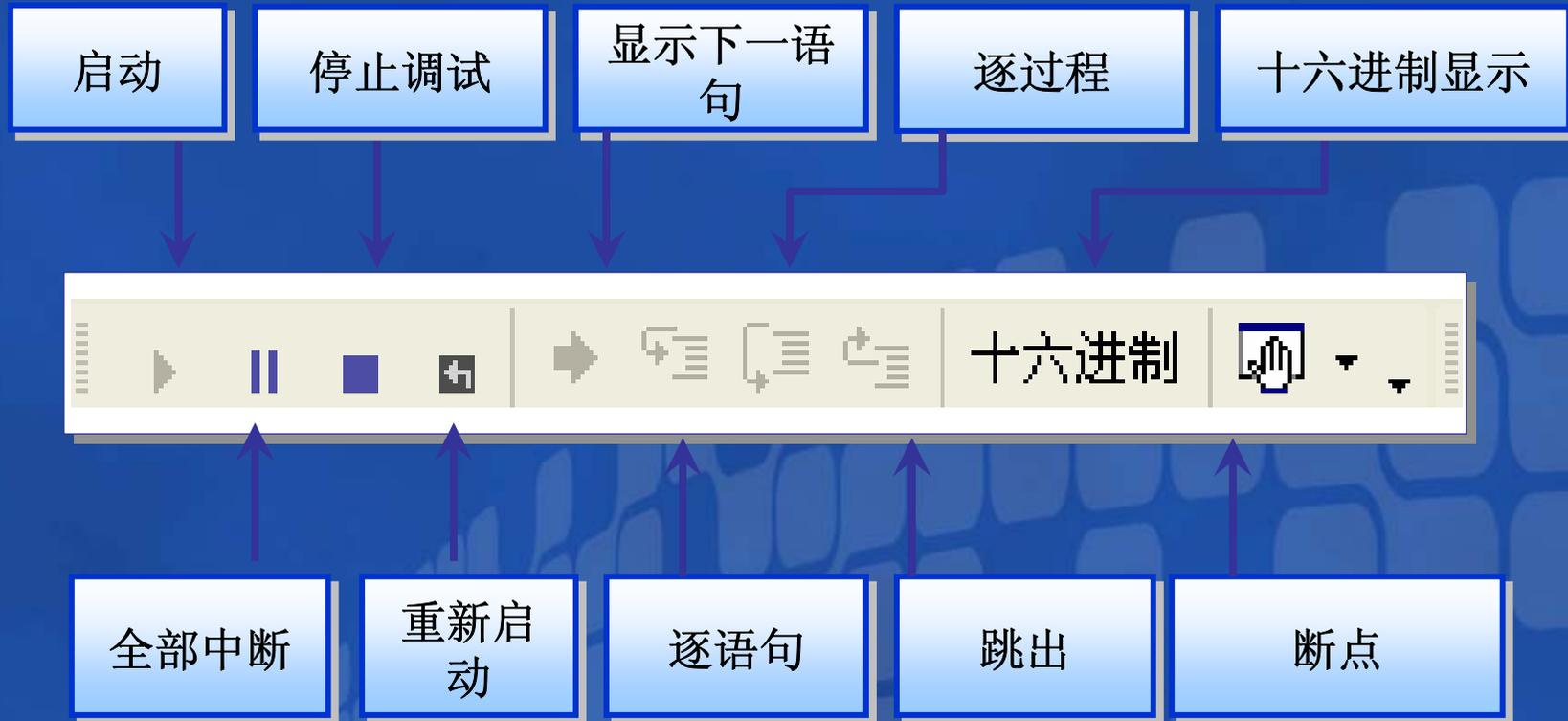
命中次数 (H)... 总是中断

确定 取消 帮助

条件
属性

命中次数
属性

调试工具栏



单步执行代码

- 逐语句或逐过程, 如果下一条代码包含对过程的调用,
 - 逐语句: 只执行调用语句, 然后停在该过程的第一行代码处
 - 逐过程: 执行整个过程, 然后停在该过程后的第一行代码处
- 跳出: 不中断地执行整个过程, 然后在调用过程的返回点处中断
- 运行到光标处: 通知调试器运行应用程序直到遇到预设的光标位置

使用调试窗口

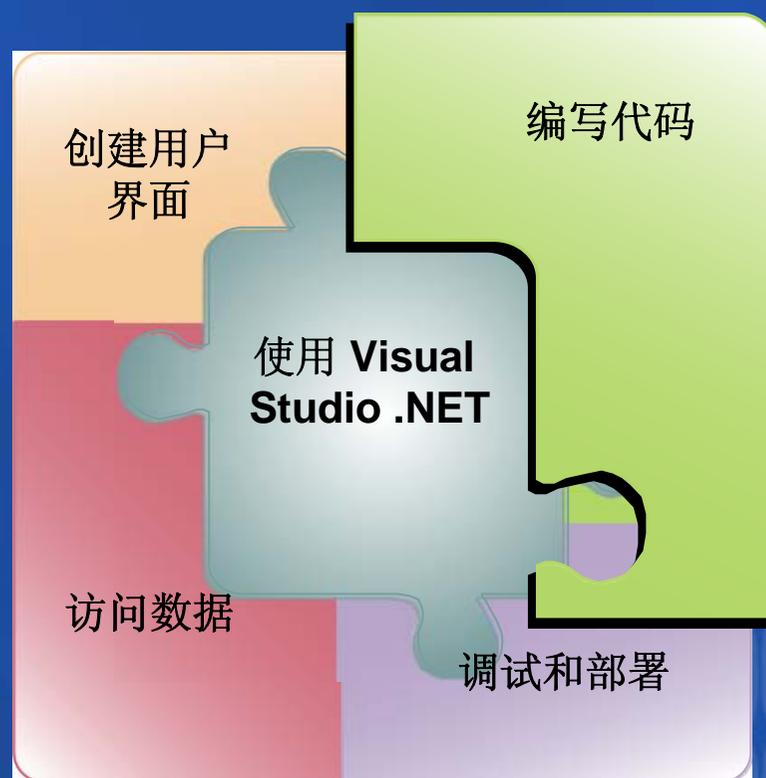
窗口	该窗口可用于...
自动	查看当前语句及其前三句和后三句语句中的变量
调用堆栈	查看正被调试的源代码的调用堆栈
局部变量	查看和修改局部变量
监视	<ul style="list-style-type: none">■ 创建要监控的自定义变量或表达式的列表■ 查看和操作被监视的表达式

使用命令窗口

- 命令窗口用来
 - 发布命令（Command 模式）
 - 调试和计算表达式（Immediate 模式）

任 务	解 决 方 案	示 例
计算表达式	在表达式前加上问号 (?)	?myVariable
从 Command 模式转到 Immediate 模式	在窗口中输入 immed，不需要大于号 (>)	immed
从 Immediate 模式转到 Command 模式	在窗口中输入 >cmd	>cmd
在 Immediate 模式下临时进入 Command 模式（以执行单条命令）	在窗口中输入命令，并在前面加上大于号 (>)	>alias

处理错误和异常



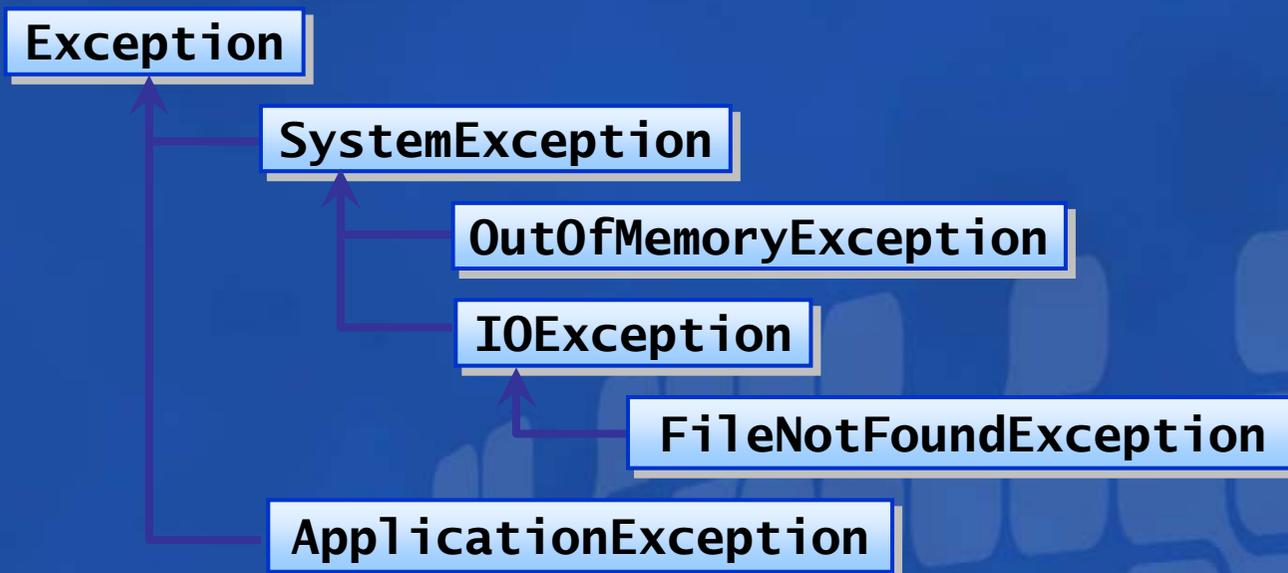
- 错误类型
- 使用调试器
- 处理异常

处理异常

- Exception 类
- 结构化异常处理
- 使用 Try...Catch 语句
- 使用 Finally 语句块
- 抛出异常
- 使用结构化异常处理的指导方针

Exception 类

- .NET 框架提供了下列异常对象模型



- 异常类允许你检索任何你所碰到的异常信息
- 异常基类的属性允许你分析异常
 - 主要的属性: **StackTrace**、**Message**、**HelpLink**、**Source**

结构化异常处理

- 检测及响应程序运行时出现的异常
- 使用 Try...Catch...Finally 语句, 可以保护可能引发错误的代码块
 - 每一个代码块都有一个或多个相关联的异常处理
 - 每一个异常处理都对其处理的异常类型指定过滤条件
- 优点
 - 可以将逻辑代码和异常处理代码隔离
 - 使得代码更加易读, 易调试

使用 Try...Catch 语句

- 有可能抛出异常的代码放入一个 Try 模块
- 在一个独立出的 Catch 块中处理异常

Try

```
fs = New FileStream("data.txt", _  
    FileMode.Open)
```

```
Catch ex As FileNotFoundException  
    MessageBox.Show("File not found")
```

```
Catch ex As Exception  
    MessageBox.Show(ex.Message)
```

End Try

程序逻辑

处理异常

使用 **Finally** 语句块

- 可选部分：如果包含，总是执行
- 在 **Finally** 块中放置清除代码，比如关闭文件

Try

```
fs = New FileStream("data.txt", FileMode.Open)
```

Catch ex As FileNotFoundException

```
MessageBox.Show("Data File Missing")
```

Catch ex As Exception

```
MessageBox.Show(ex.Message)
```

Finally

```
If Not (fs Is Nothing) Then fs.Close( )
```

End Try

抛出异常

- 用结构化异常处理语句来处理 Throw 语句抛出的异常

```
If (day < 1) Or (day > 365) Then  
    Throw New  
        ArgumentOutOfRangeException( )  
Else  
    ...  
End If
```

使用结构化异常处理的指导方针

不要对常规的错误使用结构化异常处理，使用判断结构来处理这些错误

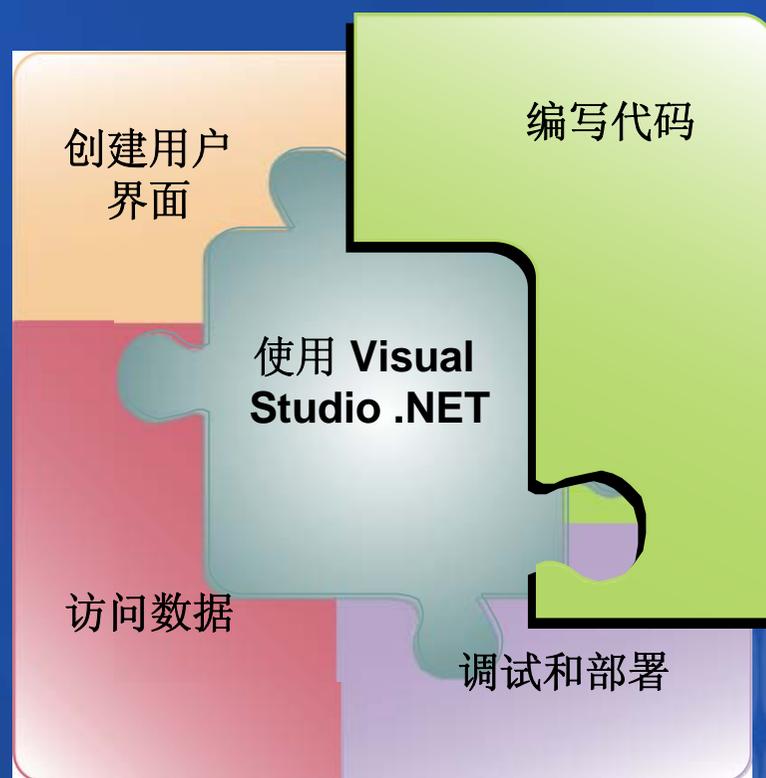
- 如 `If...End If` 语句等

对常见错误应该返回错误代码

- 示例：文件读到末尾时并不抛出 `End-Of-File` 异常

将 `Catch` 语句块从处理特殊情况排列到处理一般情况

回顾



学习完本章后, 将能够

- 定义语法错误、运行时错误以及逻辑错误并举例
- 使用 Visual Studio .NET 调试器调试代码
- 在应用程序中实现结构化异常处理

您的潜力, 我们的动力

Microsoft[®]
微软(中国)有限公司

概要

- 类的设计
- 面向对象编程的概念
- 面向对象编程的高级概念
- 使用Visio

您的潜力，我们的动力

Microsoft[®]
微软(中国)有限公司

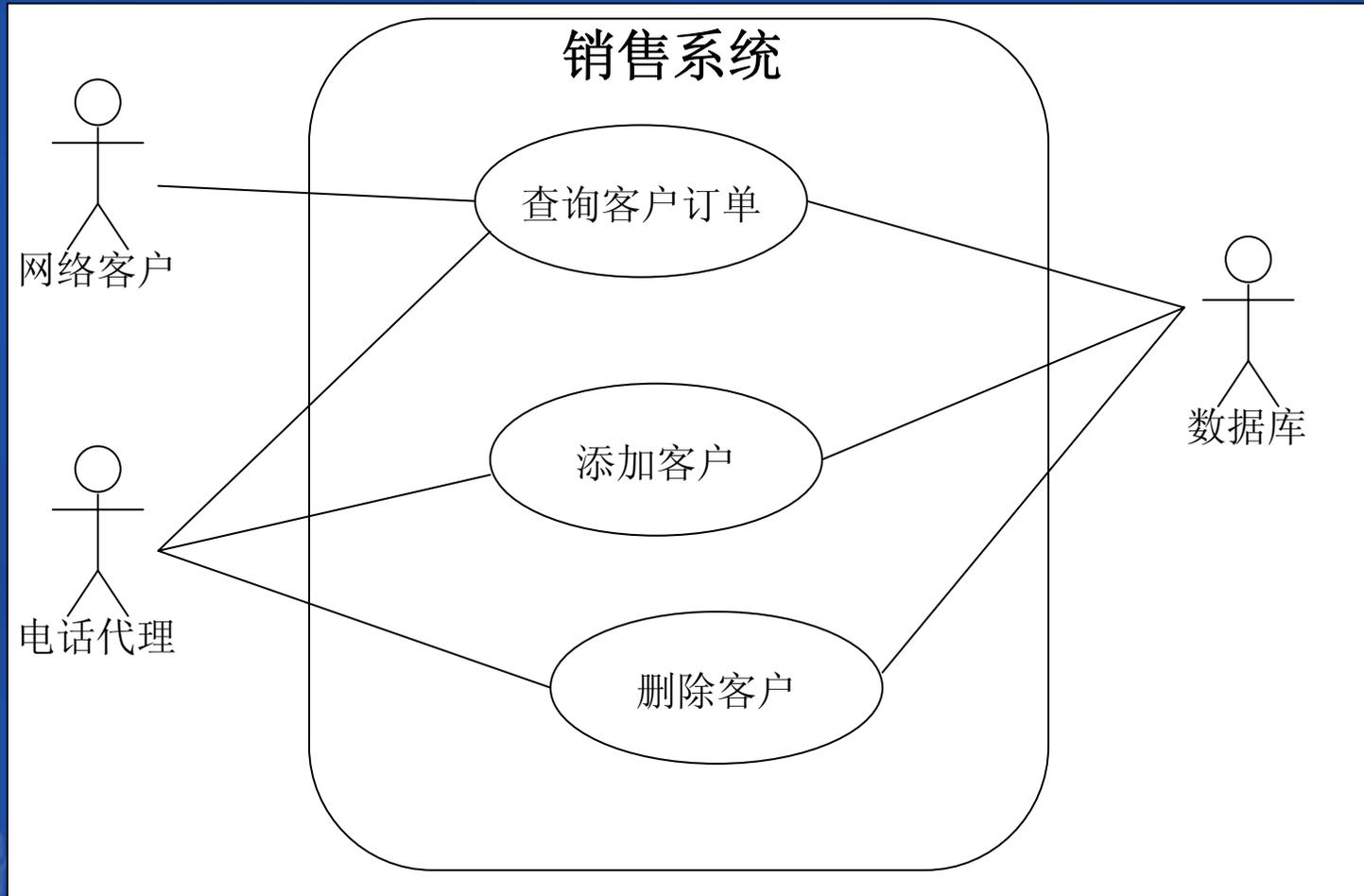
类的设计

- 用例图
- 用例图例子
- 用例描述
- 扩展用例
- 将用例转换到类

用例图

- 用例
 - 对主要的业务流程，提供一个功能性的描述
 - 使用一种非技术性的文字来描述业务流程
 - 罗列了需要解决的问题
- 角色
 - 图形化说明谁或者其它某个事务来使用这些流程

用例图例子



用例描述

- “查询客户订单”用例描述

一个用户根据特定的客户ID查看这个客户的所有订单。数据库首先验证这个ID是否有效, 如果这个客户不存在, 提示“用户不存在”。如果检索到这个ID, 则能够查看相关的客户姓名、住址和生日等相关信息, 同时也显示该客户所有订单信息, 每个订单的详情: 如订货日期、订单号、订单项都能够被查看。

用例的扩展

- 使用关系
 - 重复使用一个已经存在的用例



- 扩展
 - 改进一个用例, 使之更加精确



将用例转换成类

- 用例描述提供了一种设计类的方法
 - 名词 = 类或者属性
 - 一个用户根据特定的客户ID检索这个客户的所有订单。数据库首先验证这个ID是否有效，如果这个客户不存在，提示“用户不存在”。如果检索到这个ID，则能够查看相关的客户姓名、住址和生日等相关信息，同时也显示该客户所有订单信息，每个订单的详情：如订货日期、订单号、订单项都能够被检索。
 - 动词 = 操作（方法）
 - 例如：检索订单，验证客户ID，检索订单项

您的潜力, 我们的动力

Microsoft[®]
微软(中国)有限公司

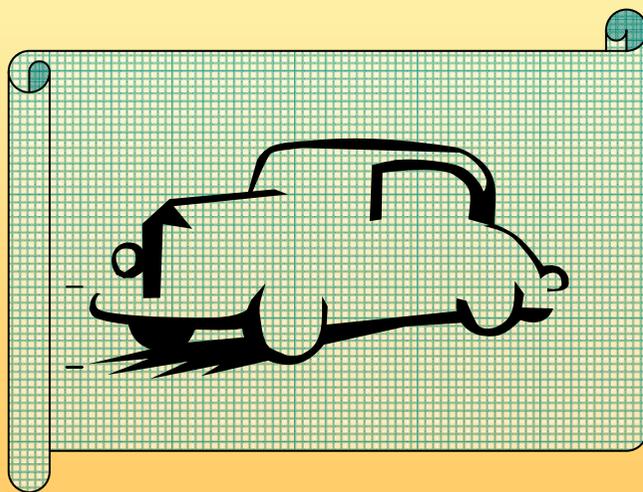
面向对象的概念

- 比较类和对象
- 封装
- 抽象
- 关联
- 聚集
- 属性和操作

比较类和对象

类

- 类是一个用来定义对象属性和操作的模板，是在设计阶段确定的。



对象

- 对象是类的一个运行时的实例。它需要内存，有生命周期。



封装

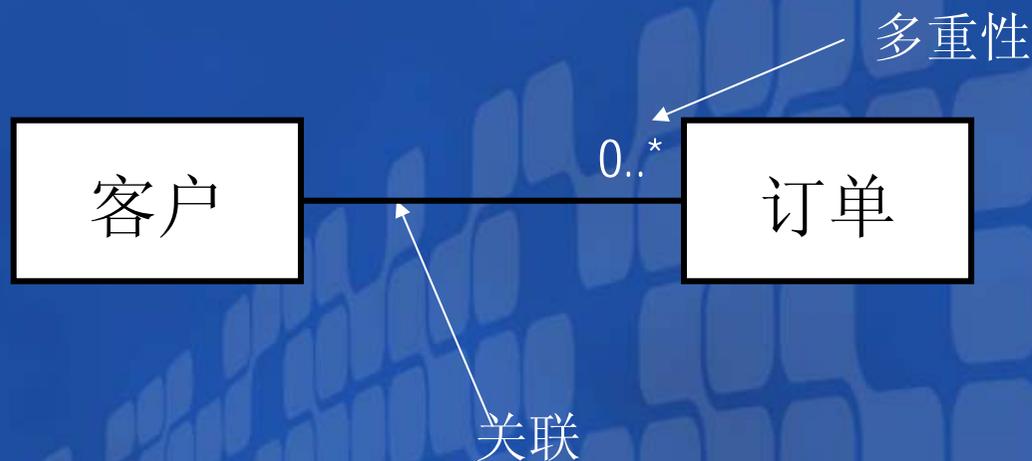
- 将一个对象如何完成其职责的细节隐藏起来，使用它的客户可以不知道这些细节
 - 客户端调用一个对象的操作、方法的时候，不需要理解这些操作、方法的工作方式等
 - 操作、方法的变化对于外部客户端来说，也是不可见的。

抽象

- 抽象?
 - 判定什么是重要的, 什么是不中要的
 - 将焦点、注意力集中到重要的事情上面
 - 忽略那些不重要的事情
 - 使用封装来加强抽象

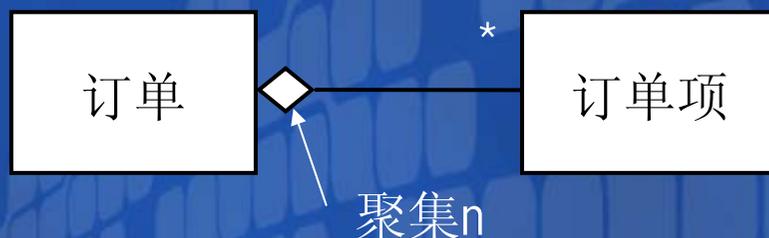
关联

- 一个类需要依赖其它类来完成某些操作
- 角色指明了关联的方向
- 多重性决定了有多少个对象能够参与这个关系



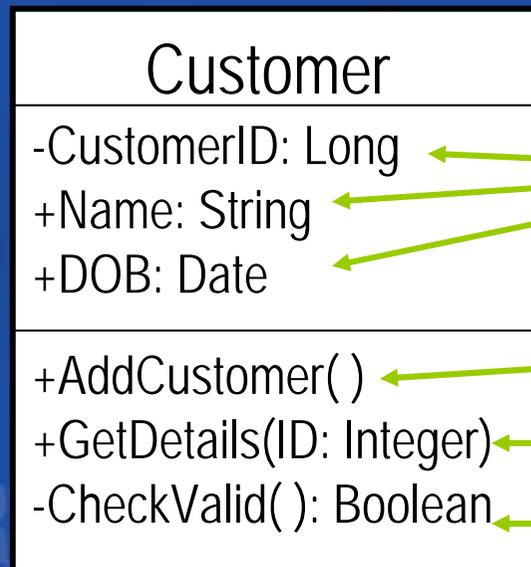
聚集

- 一个复杂类包含多个其它的类
- 一种“成员”关系
- 例如：
 - 一个订单类包含了多个订单项
 - 订单项类是订单类的成员



属性和操作

- 属性是包含在类中的数据信息
- 操作是基于这些数据完成的动作
- 可访问性：Public、Private、Protected



属性

保存操作—没有参数

初始化对象操作—整型参数

您的潜力，我们的动力

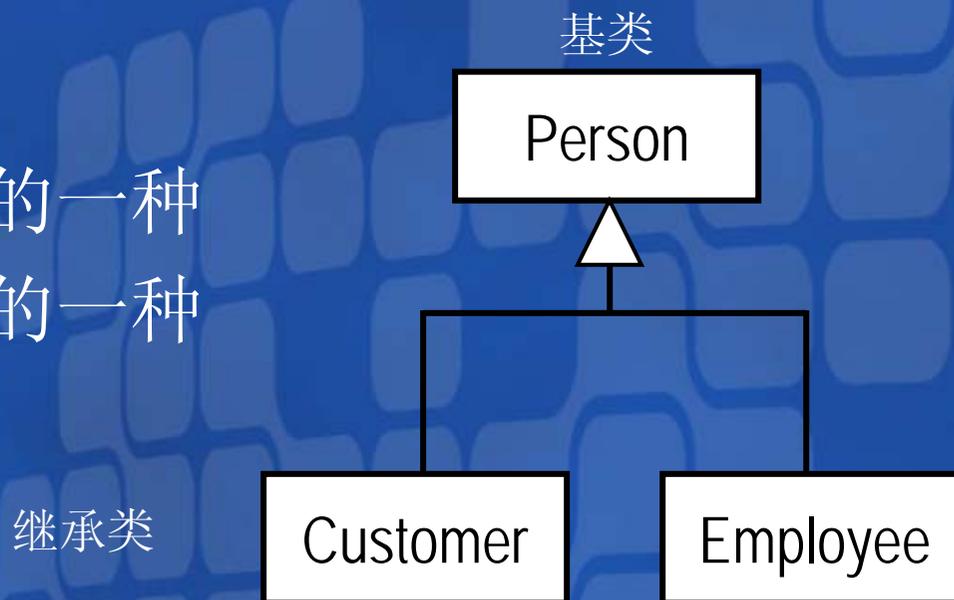
Microsoft[®]
微软(中国)有限公司

面向对象中的高级概念

- 继承
- 接口
- 多态性

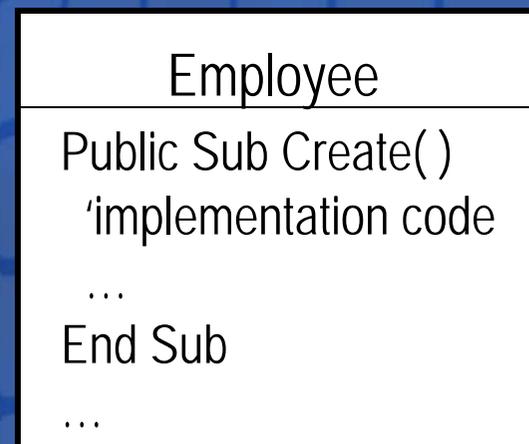
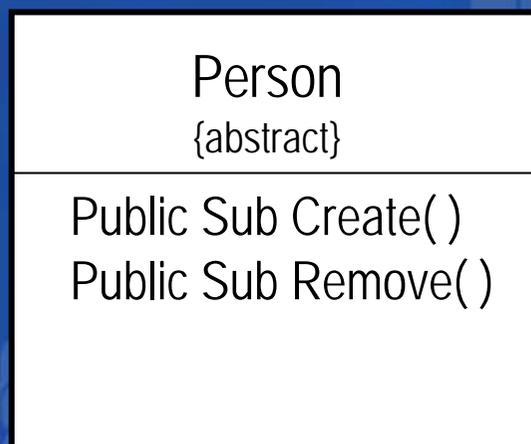
继承

- 继承表明了一种“种类”关系
- 多个类共享多个属性、操作，提高代码的重用性
- 例子：
 - 客户类是人类里面的一种
 - 员工类是人类里面的一种



接口

- 接口仅仅定义方法的签名
- 类将定义这些方法的具体执行代码
- 接口继承，是指接口是只能被继承的，不能被实际执行的代码



多态性

- 不同类的多个对象由相同的操作, 却有不同的操作行为
- 这些不同的类一般继承自同一接口
 - 类继承自接口基类
 - 每个子类都有自己的实现代码
 - 客户能像对象基类的实例一样对待这些对象, 而不需要去了解这些实际类的细节。

您的潜力，我们的动力

Microsoft[®]
微软(中国)有限公司

使用Visio

- Visio综述
- 用例图
- 类图
- 创建类图

您的潜力，我们的动力

Microsoft[®]
微软(中国)有限公司

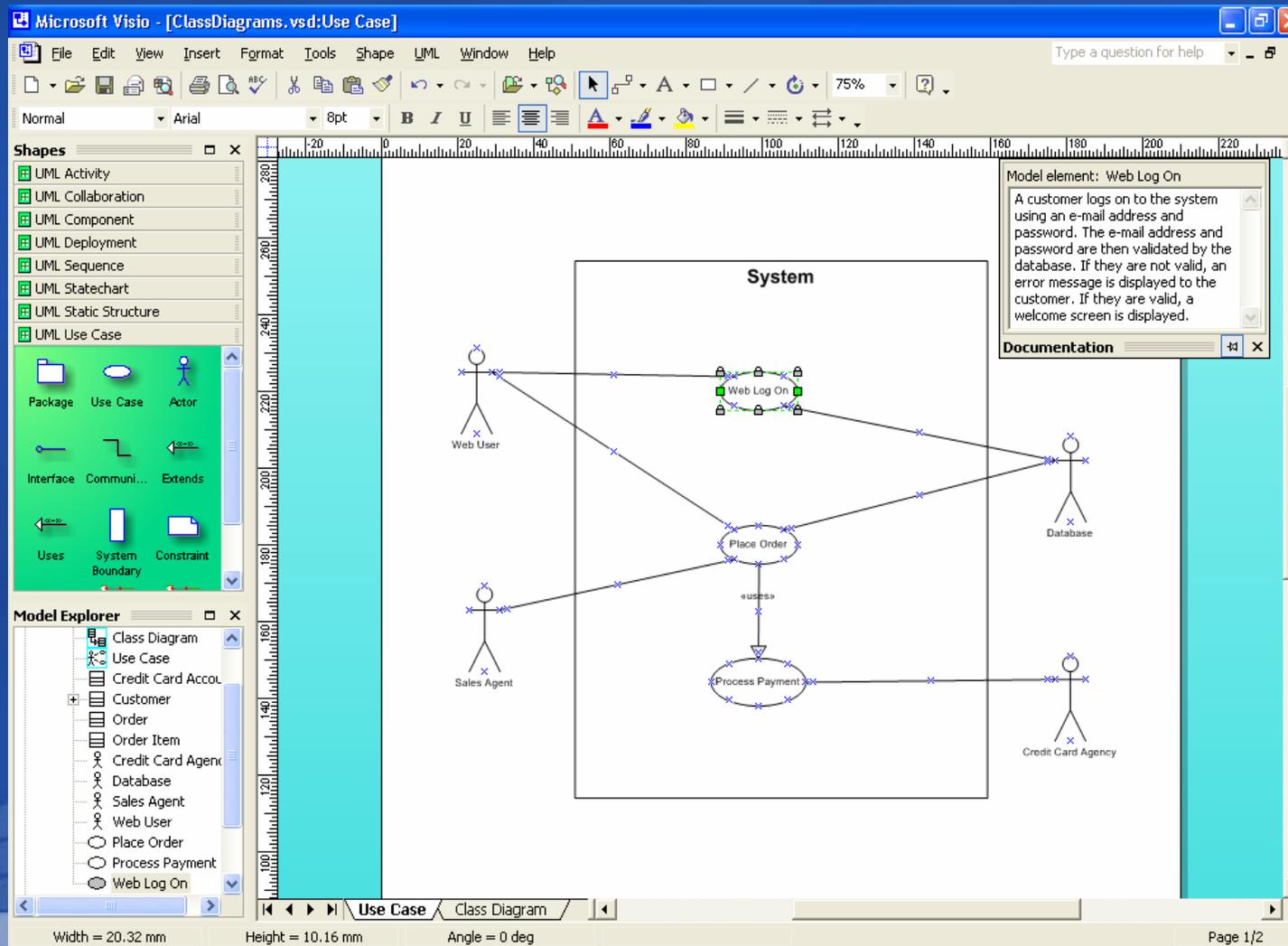
Visio综述

- 支持
 - 用例图
 - 类图、静态结构图
 - 活动图
 - 组件图
 - 部署图

用例图

您的潜力，我们的动力

Microsoft
微软(中国)有限公司



您的潜力，我们的动力

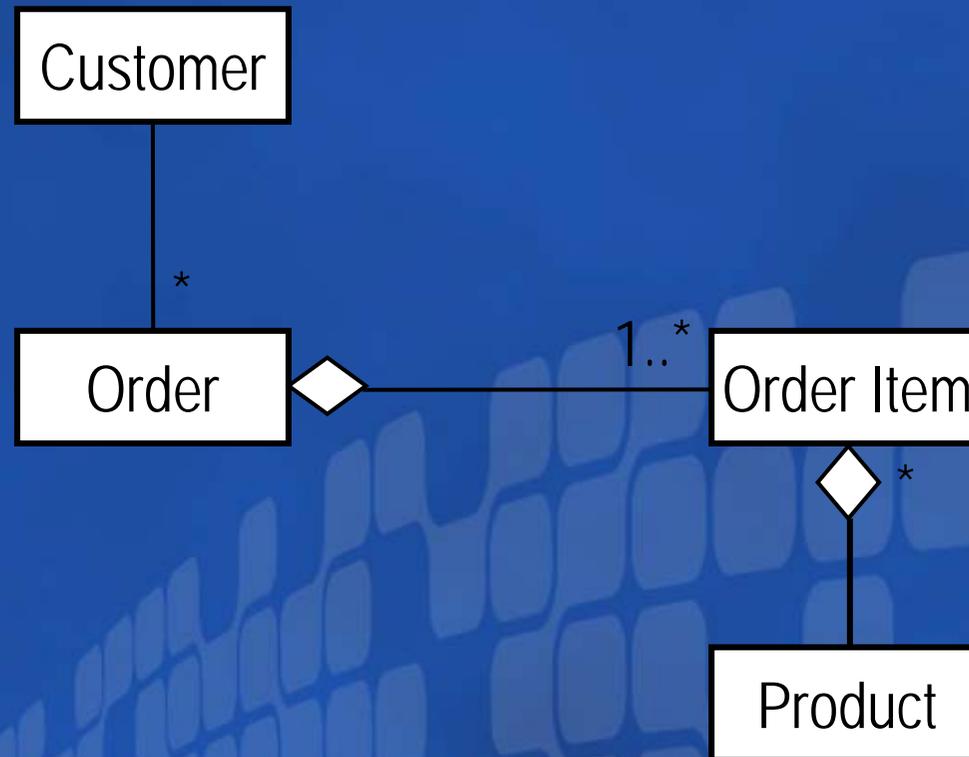
Microsoft[®]
微软(中国)有限公司

DEMO

类图

您的潜力, 我们的动力

Microsoft
微软(中国)有限公司



创建类图

您的潜力，我们的动力

Microsoft®
微软(中国)有限公司

The screenshot displays the Microsoft Visio interface for creating a UML Class Diagram. The main workspace shows a class diagram with the following classes and relationships:

- Customer**: Attributes include CustomerID (Integer), Address (String), FullName (String), Email (String), Password (String), and LogOn (in Email: String, in Password: String): Boolean.
- Credit Card Account**: Attributes include ExpiryDate (Date), BillingAddress (String), CardNumber (String), and CustomerName (String).
- Order**: Attributes include OrderID (Integer) and PlaceOrder() operation.
- Order Item**: Attributes include ProductID (Integer), Quantity (Integer), and ConfirmItem() operation.
- Product**: Attributes include Price (Double), ProductID (Integer), Category (String), Description (String), Image (String), Manufacturer (String), ProductName (String), and RetrieveDetails() operation.

Relationships shown in the diagram:

- Customer (1) is associated with Order (0..*).
- Order (1) is associated with Order Item (1..*).
- Order Item (1) is associated with Product (0..*).
- Credit Card Account (1) is associated with Customer (0..*).

The **UML Class Properties** dialog box is open for the **Customer** class, showing the following details:

- Categories:** Class
- Name:** Customer
- Full path:** UML System 1::Static Model::Top Package::Customer
- Stereotype:** (empty)
- Visibility:** public
- Documentation:** The Customer class stores information about a particular customer and allows logging on to the system.

The interface also includes a **Shapes** palette on the left with UML symbols and a **Model Explorer** on the bottom left showing the project structure.

您的潜力，我们的动力

Microsoft[®]
微软(中国)有限公司

DEMO

您的潜力, 我们的动力

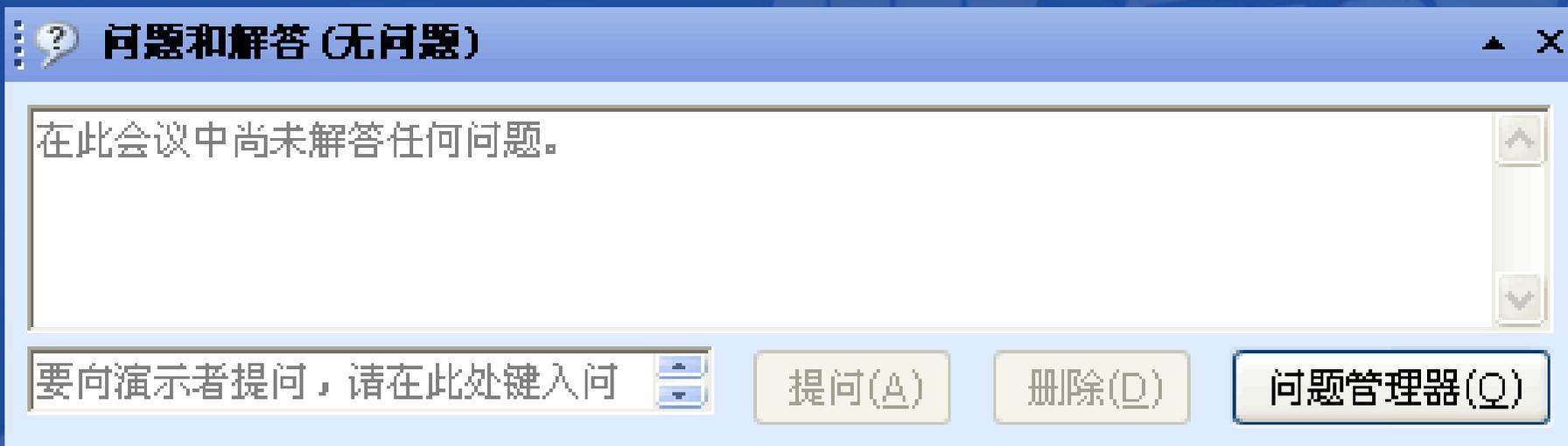
Microsoft[®]
微软(中国)有限公司

回顾

- 设计类
- 面向对象概念
- 面向对象高级概念
- 使用Visio

Q&A

如需提出问题，请单击“提问”按钮并在随后显示的浮动面板中输入问题内容。一旦完成问题输入后，请单击“提问”按钮。



The screenshot shows a window titled "问题和解答 (无问题)" (Questions and Answers (No Questions)). The window contains a text area with the message "在此会议中尚未解答任何问题。" (No questions have been answered in this meeting yet.). Below the text area are three buttons: "提问(A)" (Ask), "删除(D)" (Delete), and "问题管理器(Q)" (Question Manager). There is also a small input field with the placeholder text "要向演示者提问，请在此处键入问" (To ask a question to the presenter, enter the question here).

您的潜力，我们的动力

Microsoft[®]
微软(中国)有限公司

Microsoft[®]

msdn


MSDN Webcasts